Neurocomputing 157 (2015) 11–21

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

DLANet: A manifold-learning-based discriminative feature learning network for scene classification

Ziyong Feng [a], Lianwen Jin [a,*], Dapeng Tao [b,c], Shuangping Huang [d]

[a] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China
[b] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[c] The Chinese University of Hong Kong, Hong Kong, China
[d] College of Engineering, South China Agricultural University, Guangzhou, China

ARTICLE INFO

Article history:
Received 20 August 2014
Received in revised form
12 December 2014
Accepted 21 January 2015
Communicated by X. Gao
Available online 3 February 2015

Keywords:
Convolution neural network
Manifold learning
DLA Network
Scene classification

ABSTRACT

This paper presents Discriminative Locality Alignment Network (DLANet), a novel manifold-learning-based discriminative learnable feature, for wild scene classification. Based on a convolutional structure, DLANet learns the filters of multiple layers by applying DLA and exploits the block-wise histograms of the binary codes of feature maps to generate the local descriptors. A DLA layer maximizes the margin between the inter-class patches and minimizes the distance of the intra-class patches in the local region. In particular, we construct a two-layer DLANet by stacking two DLA layers and a feature layer. It is followed by a popular framework of scene classification, which combines Locality-constrained Linear Coding–Spatial Pyramid Matching (LLC–SPM) and linear Support Vector Machine (SVM). We evaluate DLANet on NYU Depth V1, Scene-15 and MIT Indoor-67. Experiments show that DLANet performs well on depth image. It outperforms the carefully tuned features, including SIFT and is also competitive to the other reported methods.

1. Introduction

Scene classification is an important research task in robotics and computer vision. It plays a key role for various practical applications, e.g., robotics place recognition [1], robotics path planning [2,3], semantic recognition [4], and content-based image retrieval [5].

However, scene classification is very challenging due to the wide variety of intra-class scene, sophistication of lighting and background, even different view angles. For example, a natural scene dataset collected by Vogel and Schiele [5] contains 6 categories (coast, forest, mountain, open country, river and sky/cloud). A mountain overlaid with green grass is very different from that covered by snow because of the variation of color. But the green grass overlaid mountain is easily confused with forest because of the similar color and texture. Another dataset organized by Oliva and Torralba [6] includes both the natural scenes and man-made scenes, which enrich the semantic of scene. By contrast to the outdoor scene datasets described above, the indoor scene recognition is similar to multiple objects recognition. A living room may include bed, chairs, night table and people. Furthermore, the non-rigid deformation and occlusion with these objects will be observed in an indoor scene, which increase the difficulty of recognition. Clearly, adding the 3D information can help recognizing scene [7]. Therefore, some datasets contain regular color images and the corresponding depth maps captured by Microsoft Kinect, e.g., NYU Depth V1 [7], NYU DepthV2 [8] and SUN3D [9]. Kinect uses structured light method to capture the accurate depth map of a scene, which can be aligned with the device's VGA camera easily. However, Kinect works reliably only on indoor scenes because the effective range of the depth camera does not apply to bad lighting conditions. Remarkably, since a limited number of scene categories cannot simulate the daily life, Xiao et al. [10] established a huge dataset to capture the richness and diversity of environments.

To solve this comprehensive problem, a large number of approaches have been proposed. These approaches can be divided into two categories according to the literature [11]. The one category is that using the global low level features to represent a scene image. In general, representative global color features, e.g., HSV color histogram [12], color coherence vectors [13] and color moment [14], are the most employed thanks to the invariant of scaling, rotation, perspective, and occlusion. Although color features perform better than texture and shape, texture and shape features are also used for scene classification as another cue. They can encode the edge information like the straight horizontal and vertical edges in an urban scene. However the global low level features fail to work in spite of large changes in viewing conditions, occlusions and clutters. Normally only a small number of scene categories problem use global low level features [15].

The other category represent scene images associated with detected interest points (or image blocks) based on some descriptors.

* Corresponding author. Tel.: +86 20 87113540
E-mail address: lianwen.jin@gmail.com (L. Jin).

http://dx.doi.org/10.1016/j.neucom.2015.01.043
0925-2312/

Then a codebook is constructed by these descriptors obtained from training images. Thus an image is represented by a histogram of the codebook, which transfers low level features to high level features, also called coding. The Bag of Words (BOW) model is a classical coding method for scene classification. The BOW model and its variations can provide the more representative features for the images. Instead of using BOW model, sparse coding is employed as the coding method which has lower reconstruction error and sparse representation. Generally the codes from an image would be pooled to form an image feature. Finally, the features are used to train a classifier for scene classification. However the low level features play an important role in the classification system. Lots of efforts have been made to design low level features for classification tasks at hand.

In the past few years, Deep Neural Networks (DNNs) have received intensive attentions, because they can automatically and simultaneously discover low level and high level features, and achieved astonishing results on various databases. For example, a deep neural networks conducted by stacking Restricted Boltzmann Machines (RBMs) [40] and regularized autoencoders [44] perform much better than traditional neural networks. Moreover, for considering the topological structure, Convolutional Neural Networks (CNNs) [45] constructed by convolutional and pooling operation is more suitable for computer vision tasks. Therefore, on various database, CNN obtained the-state-of-art performance [53–55]. But there are lots of parameters in a deep CNN to be tune given the enormous data, which leads to high computation even using extremely fast GPU implementation on GPU [52].

To design a simple deep learning network, Chan et al. [56] proposed a convolutional neural network without active function and pooling layers. Instead of BP algorithm, they adopted PCA or LDA to learn the bases and treat the bases as filters in CNN, called PCANet and LDANet, respectively. In the output stage, binary quantization and block-wise histogram operator of the binary codes are employed to generate the output features. The experiments show that a two-layer PCANet is superior to the state-of-the-art features for some image classification tasks.

In this paper, we deploy the structure of PCANet to learn the local features but explore filters learnt by Discriminative Locality Alignment (DLA) [57], which can project the patches closer intra-class and further otherwise. It is found that DLA can cope with the nonlinearity of the distribution of samples while preserving the discriminative information and enhance the importance of marginal samples for discriminative subspace selection [57]. The advantages of DLA will benefit the classification tasks. We train the filters in CNN with the manifold assumption and it is expected that the features learnt by DLA Network (DLANet) contain more effective discriminative information. Then the features computed by the learnt DLANet are fed into LLC–SPM to represent the images. Eventually, to classify scenes, we utilize linear SVM because it is more suitable for LLC. To evaluate the effectiveness of the proposed DLANet feature, we compare it with other scene classification algorithm on NYU Depth V1 [7], Scene-15 [25] and MIT Indoor-67 [26]. Compare with the classification system using PCANet/LDANet in [65], we learn the filters by DLA and use the DLANet features to generate a more efficient image representation with coding and spatial pooling. The flow diagram of our classification system is illustrated in Fig. 1. The main contribution of this paper is the newly developed DLANet feature learning algorithm, a novel manifold-based discriminative feature learning algorithm, for scene classification.

The rest of the paper is organized as follows. In Section 2, we review related work on feature learning. Then we introduce the proposed DLANet feature learning algorithm in detail in Section 3. Section 4 shows the experimental results on the NYU Depth V1, Scene-15 and MIT Indoor-67. We conclude the paper in Section 5.

2. Related work

2.1. Low level local features

Features play an essential role in classification tasks. Numerous efforts have been made to design low level features for classification tasks at hand. Some hand-crafted features for scene classification will be introduced in this section. Scale Invariant Feature Transform (SIFT) [34] feature and descriptor are popularized in the computer vision community, which are originally designed for recognizing the same object appearing under different conditions. Because of the high discriminative power, SIFT is adopted for scene classification widely 35][17,22,33,35]. The Histograms of Oriented Edges (HOG) [36] descriptor is widely used for pedestrian and object detection. A variant of HOG [37] computes instead both directed and undirected gradients as well as a four dimensional texture-energy feature, but projects the feature onto a 31-dimensional space. The experimental results [10] show that this variant of HOG gains the decent performances among various features. For recognizing topological places and scene classification, Wu et al. [1] introduced Census Transform Histogram (CENTRIST) which has several important advantages, e.g., no parameter to tune, extremely fast, and easy to implement. However the above low level features are designed manually for some specific data and tasks.

Designing effective features for new data and tasks usually requires new domain knowledge and the original features are not suitable for new problems. For instance, the depth images acquired by the Kinect sensor receive intensive attentions. It is unclear how this depth information can be exploited using existing features. Naturally a depth image is treated as a gray level intensity image and existing features e.g. SIFT [35] are applied. Histogram of Oriented Normal Vectors (HONV) [38] is designed specifically to capture the local 3D geometric characteristics for the purpose of object recognition in a depth image.

Motivated by kernel-based feature learning, Bo et al. [39] present a set of kernel features on depth images that describe size, shape and edges in a unified framework. The kernel features measure the local descriptors in kernel space instead of original linear space. The kernel descriptors constructed by projecting the infinite-dimensional feature vectors to the learned basis vectors have more appropriate similarity measure for local patches and significantly outperform the hand-crafted features.

However, designing effective features for new data and tasks usually requires domain knowledge. Because of the limitation of hand-crafted features, learning features from data becomes hot. In recent years, DNN becomes a powerful tool to learn features, in

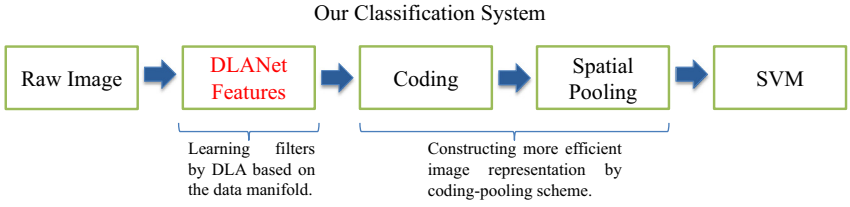

Fig. 1. The flow diagram of our classification system.

which layer-wise unsupervised pretraining is key stage for initializing a DNN. This pretraining procedure can be deemed as using the unsupervised feature learning to learn a new transformation to extract effective features. For example, Restricted Boltzmann Machines (RBMs) [40] minimize the energy function, which are trained by the Contract Diversity (CD) [41] algorithm in an approximate way. Autoencoder [42] jointly defines an encoding function and the corresponding decoding function, which are simultaneously trained by minimizing the reconstruction error. The encoding function is regarded as feature-extracting function. Recently variants of regularized autoencoders are developed to improve the generalization, e.g., Contractive Autoencoders (CAEs) [43] and Denoising Autoencoders (DAEs) [44].

Although these methods achieve promising results, they ignore the topological structure of the input data in computer vision tasks. Convolutional Neural Networks (CNNs) [45] are popular, because they define local receptive fields [46] so that each low-level feature will be computed from only a subset of the input. The convolutional operation ensures that an output unit only associates with the input units in a receptive field. Commonly, it is followed by the max-pooling, which can pool feature layer some degree of invariance to input translations. However the typical CNN is trained in the supervised manner. Hence to train a convolutional layer in an unsupervised fashion, an intuitive method is that collecting the patches from training set randomly and applying several feature learners [47]. Coates and Ng [47] found that simple *k*-means clustering is superior to many sophisticated feature learners. Convolutional versions of RBM [48,49] were proposed to directly train the entire convolutional layers utilizing an unsupervised criterion. Several methods combine the sparse coding with CNN, e.g., Predictive Sparse Decomposition (PSD) [50] and Deconvolutional Networks [51].

To design a simple DNN, Chan et al. [56] presented PCA Network and its variations, LDA Network and random network. They apply PCA or LDA to learn the bases and treat the bases as filters in CNN. PCA calculates the principal components, which are a subset of the orthonormal bases carrying the most principal energy and projects the data points onto the subspace spanned by principal components. PCA is suitable for reconstruction of Gaussian distributed data but not for classification. On the other hand, LDA is a linear dimension reduction algorithm that can make use of the discriminative information. It tries to maximize the distance of inter-class data and minimize the distance of intra-class data in a low dimension space. However LDA cannot discover the nonlinear structure hidden in the high dimensional non-Gaussian distributed data and assume that each sample makes an equivalent contribution to discriminative dimension reduction. Manifold learning algorithms efficiently reduce the dimensionality. For example, Locally Linear Embedding (LLE) [58] finds a low-dimensional, neighborhood-preserving embedding of the high-dimensional data in an unsupervised fashion. ISOMap [59] considers the geodesic distance between samples, which can discover the nonlinearity of the high dimensional data. Discriminative Locality Alignment (DLA) [57] projects the patches closer intra-class and further otherwise, which is superior to LLE and ISOMap for classification tasks because it considers the discriminative information. Patch Alignment Framework (PAF) [60] unifies the existing dimension reduction algorithms.

Many DLA variants are proposed in recent years. Guan et al. [61] presented Non-negative Discriminative Locality Alignment (NDLA) which is incorporated by non-negative constraints on both the bases and the coordinates. Manifold Elastic Net (MEN) [62] expects to minimize the classification errors explicitly and obtains a sparse projection matrix by adding the lasso penalty. To extend the original DLA to tensor space, Mu et al. [63] proposed the three-way DLA (TWDLA) for C1 third-order tensor feature. Zhang et al. [64] also presented tensor extension of conventional DLA (TDLA) for hyperspectral image spectral-spatial feature extraction.

2.2. Features coding

The coding process transfers low level features to high level features. The typically encoding is based on Vector Quantization (VQ), hard assigning each descriptor to the closest codeword in the codebook learning by a clustering algorithm such as k-means. Therefore the hard assignment is a coarse estimation of the descriptor distribution. Soft assignment is a method that assigns one descriptor to several codewords proposed to achieve sophisticated estimation and less information loss. Kernel codebook [16] is a soft assignment encoding which estimates the distribution by kernel density to allow a degree of ambiguity in assigning codewords to descriptors. Yang *et al.* [17] replaced the VQ with Sparse Coding (SC) which can obtain nonlinear codes. Yu *et al.* [18] empirically found that SC results tend to be local – active coefficients are often assigned to codewords close to the descriptor encoded. Hence they presented Local Coordinate Coding (LCC) [18] modified by SC, which explicitly encourages the coding to be local, and pointed out that under manifold assumptions locality is more important than sparsity in practice, for successful nonlinear function learning using the learned codebook. LCC which is similar to SC should optimize a weighted LASSO, which is computational expensive. Thus a variation of LCC, called Locality-constrained Linear Coding (LLC) [19], project each descriptor into its local-coordinate system with lower computational complexity because it has a closed form solution. Combined with a linear Support Vector Machine (SVM), LLC achieves encouraging accuracy for image classification. In addition, they utilized K-nearest-neighbor (K-NN) search and constrained least squares fitting to approximate the solution of LLC, which further reduces the computational complexity. Alternatively Fisher Vectors (FV) [20,21], an extension of the VQ, encode the average first and second order differences between the descriptors and the centers of a Gaussian Mixture Model (GMM). Thanks to the finer estimation, FV performs better than other encoding technique [22]. Recently supervised encoding methods reported that learning a discriminative codebook improves the classification performance [23,24].

2.3. Spatial pooling

Spatial Pyramid Matching (SPM) [25] is employed to represent the whole image for the subsequent recognition. Because SPM encodes the descriptor spatial layout, it has been widely utilized in the recent state-of-the-art image classification systems [17,22]. The image is partitioned into increasingly finer spatial sub-regions and computes histograms of local features from each sub-region. Empirically, 1×1, 2×2, and 4×4 sub-regions (typical Spatial Pyramid) are used in the Caltech-101 data. Another partition scheme is 1×1, 2×2, and 3×1 sub-regions, which is suitable for the images with "sky" on top and/or "ground" on bottom. To pool the codes in the sub-regions, various pooling functions, e.g., average-pooling, max-pooling [17,19], Geometric ℓ_p-norm Pooling (GLP) [27], or Geometric Phrase Pooling (GPP) [32], are introduced. If VQ codes are employed, average-pooling amounts to the histogram of each sub-region. However, SC and LLC prefer max-pooling inspired by the visual cortex (V1). Boureau et al. [28] theoretically analyzed the pooling methods and discussed the influence of average-pooling and max-pooling on the different encoding schemes. How to partition an image is based on the priority empirically and the partition scheme is fixed on a database. Recently a few works [29,30] attempt to learn and design the pooling regions, which also improve the classification performance but increase computational complexity. Lin et al. [31] proposed a model that can learn important spatial pooling regions (ISPRs) and discriminative part appearance together. Compared to traditional SPM, Orientational Pyramid Matching (OPM) [33] uses the 3D orientations to index the image blocks and form the pyramid in the orientational space. The experiments show that OPM is an effective complement of SPM.

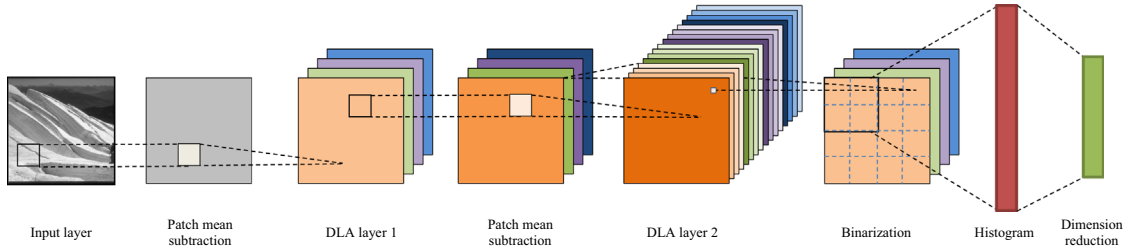

Fig. 2. The illustration of two-layered DLANet feature learning scheme. In the first DLA layer, a gray scale image is convolved with four filters (four colors) learnt by DLA. In the second DLA layer, each feature map is convolved with four filters (four colors from dark to light). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. DLANet feature learning

In this section, we introduce a novel manifold-learning-based discriminative feature learning algorithm, DLANet. The diagram of the proposed DLANet is shown in Fig. 2.

DLANet adopts the structure of PCANet that learns the convolutional filter bank through PCA [56]. However, the shortcomings of PCA will affect the performance of the network. To avoid these disadvantages and improve the network, DLA is utilized to construct the DLANet in this paper. Therefore we apply it to learn the filter bank in the network and hope the more effective features can be discovered automatically.

3.1. The first DLA layer

Given N training scene images $\{\mathbf{X}_i\}_{i=1}^{N}$ of size $m \times n$, each image has the corresponding class label $y_i = 1, 2, \ldots, k$, where k is the category number. We successively take image blocks of size $l \times l$ of each image and then each block will be vectorized. For the ith image, we have data matrix $\mathbf{P}_i = (\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \ldots, \mathbf{p}_{i,mn}) \in \mathbb{R}^{l^2 \times mn}$, where $\mathbf{p}_{i,j}$ is the jth vectorized block. For normalization, each block will subtract its mean and then we obtain the normalized data matrix:

$$\overline{\mathbf{P}}_i = (\overline{\mathbf{P}}_{i,1}, \overline{\mathbf{P}}_{i,2}, \ldots, \overline{\mathbf{P}}_{i,mn}), \tag{1}$$

where $\overline{\mathbf{p}}_{i,j}$ is the normalized block which has zero mean and its class label is the same as that of the corresponding image, i.e., $\mathbf{y}_i = y_i 1_{mn}$, where $1_{mn} = (1, \ldots, 1)^T \in \mathbb{R}^{mn}$. For all training images, we concatenate their corresponding normalized data matrices to a large matrix:

$$\mathbf{P} = (\overline{\mathbf{P}}_1, \overline{\mathbf{P}}_2, \ldots, \overline{\mathbf{P}}_N) \in \mathbb{R}^{l^2 \times Nmn}. \tag{2}$$

Thus we have Nmn samples of l^2 dimensionality. For convenient description, we rewrite it as the concatenation of vectors with successive index, i.e., $\mathbf{P} = (\overline{\mathbf{p}}_1, \overline{\mathbf{p}}_2, \ldots, \overline{\mathbf{p}}_{Nmn})$. We want to find a projection matrix $\mathbf{U} \in \mathbb{R}^{l^2 \times D_1}$ to linearly map samples from the high-dimensional space $\mathbb{R}^{l^2}$ to a low-dimensional subspace $\mathbb{R}^{D_1}$, with $D_1 < l^2$, where D_i is the dimensionality to be reduce to in layer i. It is noted that D_i is also the number of filters in layer i and because of the constraint $D_i < l^2$, we cannot set the number of filters in layer i larger than the block size. Based on the above discussions, we employ DLA to find the projection matrix $\mathbf{U}$. We illustrate the feature learning procedure in Fig. 3.

For the ith sample $\overline{\mathbf{p}}_i$, $i = 1, 2, \ldots, Nmn$, we can categorize the samples into the intra-class samples and inter-class samples. The k_1 closest intra-class samples $\overline{\mathbf{p}}_{i^1}, \overline{\mathbf{p}}_{i^2}, \ldots, \overline{\mathbf{p}}_{i^{k_1}}$, the k_2 closest inter-class samples $\overline{\mathbf{p}}_{i_1}, \overline{\mathbf{p}}_{i_2}, \ldots, \overline{\mathbf{p}}_{i_{k_2}}$ and the given sample $\overline{\mathbf{p}}_i$ form a subset of the entire sample set, which is

$$\hat{\mathbf{P}}_i = \left(\overline{\mathbf{p}}_i, \overline{\mathbf{p}}_{i^1}, \overline{\mathbf{p}}_{i^2}, \ldots, \overline{\mathbf{p}}_{i^{k_1}}, \overline{\mathbf{p}}_{i_1}, \overline{\mathbf{p}}_{i_2}, \ldots, \overline{\mathbf{p}}_{i_{k_2}}\right) \in \mathbb{R}^{l^2 \times (k_1 + k_2 + 1)}. \tag{3}$$

The corresponding low-dimensional representation of $\hat{\mathbf{P}}_i$ after transformation is $\mathbf{Z}_i = \left(\mathbf{z}_i, \mathbf{z}_{i^1}, \mathbf{z}_{i^2}, \ldots, \mathbf{z}_{i^{k_1}}, \mathbf{z}_{i_1}, \mathbf{z}_{i_2}, \ldots, \mathbf{z}_{i_{k_2}}\right) \in \mathbb{R}^{D_1 \times (k_1 + k_2 + 1)}$. The corresponding index set is defined as $S_i = \{i, i^1, i^2, \ldots, i^{k_1}, i_1, i_2, \ldots, i_{k_2}\}$. To minimize the distance of the intra-class samples $\mathbf{z}_i, \mathbf{z}_{i^1}, \mathbf{z}_{i^2}, \ldots, \mathbf{z}_{i^{k_1}}$ in the low dimensional space, we expect that distances between $\mathbf{z}_i$ and intra-class neighbor samples $\mathbf{z}_{i^1}, \mathbf{z}_{i^2}, \ldots, \mathbf{z}_{i^{k_1}}$ are as small as possible, so the distance is

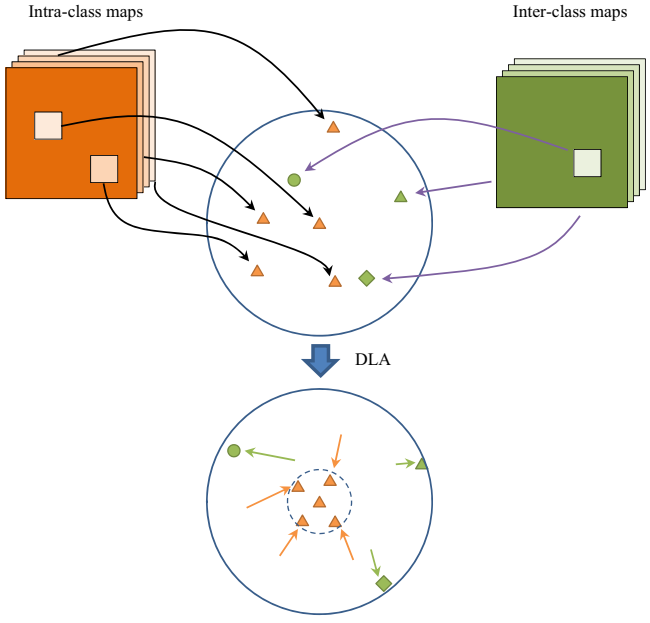

Fig. 3. Learning DLA filters from neighbor patches. Orange triangles denote the intra-class samples and green shapes denote the inter-class samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$M_1(\mathbf{z}_i) = \sum_{j=1}^{k_1} ||\mathbf{z}_i - \mathbf{z}_{i^j}||^2. \tag{4}$$

Simultaneously, we expect that distances between $\mathbf{z}_i$ and inter-class neighbor samples $\mathbf{z}_{i_1}, \mathbf{z}_{i_2}, \ldots, \mathbf{z}_{i_{k_2}}$ are as small as possible, the distance is

$$M_2(\mathbf{z}_i) = \sum_{p=1}^{k_2} ||\mathbf{z}_i - \mathbf{z}_{i_p}||^2. \tag{5}$$

Based on the manifold assumption [57], the part discriminator can be obtained by linearly combining (4) and (5):

$$\begin{aligned} &\operatorname*{argmin}_{\mathbf{z}_i} \left(M_1(\mathbf{z}_i) - \gamma M_2(\mathbf{z}_i)\right) \\ &= \operatorname*{argmin}_{\mathbf{z}_i} \left(\sum_{j=1}^{k_1} ||\mathbf{z}_i - \mathbf{z}_{i^j}||^2 - \gamma \sum_{p=1}^{k_2} ||\mathbf{z}_i - \mathbf{z}_{i_p}||^2\right), \end{aligned} \tag{6}$$

where $\gamma \in [0, 1]$ is a tuning parameter to balance the contributions of M_1 and M_2. To simplify the part objective function (6), we define

the coefficients vector:

$$\mathbf{w}_i = \left(\overbrace{1,\ldots,1}^{k_1}, \overbrace{-\gamma,\ldots,-\gamma}^{k_2} \right)^T, \tag{7}$$

then the part objective function (6) transforms to

$$\begin{aligned}
&\underset{\mathbf{z}_i}{\operatorname{argmin}} \left(\sum_{j=1}^{k_1} ||\mathbf{z}_i - \mathbf{z}_{i_j}||^2 (\mathbf{w}_i)_j + \sum_{p=1}^{k_2} ||\mathbf{z}_i - \mathbf{z}_{i_p}||^2 (\mathbf{w}_i)_{k_1+p} \right) \\
&= \underset{\mathbf{z}_i}{\operatorname{argmin}} \left(\sum_{j=1}^{k_1+k_2} ||\mathbf{z}_{S_i\{1\}} - \mathbf{z}_{S_i\{j+1\}}||^2 (\mathbf{w}_i)_j \right) \\
&= \underset{\mathbf{z}_i}{\operatorname{argmin}} \operatorname{tr} \left(\mathbf{Z}_i \begin{pmatrix} -\mathbf{1}_{k_1+k_2}^T \\ \mathbf{I}_{k_1+k_2} \end{pmatrix} \operatorname{diag}(\mathbf{w}_i) (-\mathbf{1}_{k_1+k_2}, \mathbf{I}_{k_1+k_2}) \mathbf{Z}_i^T \right) \\
&= \underset{\mathbf{Z}_i}{\operatorname{argmin}} \operatorname{tr} \left(\mathbf{Z}_i \mathbf{L}_i \mathbf{Z}_i^T \right),
\end{aligned} \tag{8}$$

where $\mathbf{1}_{k_1+k_2} = (1,\ldots,1)^T \in \mathbb{R}^{k_1+k_2}$, $\mathbf{I}_{k_1+k_2} = \operatorname{diag}(\mathbf{1}_{k_1+k_2})$ is the identity matrix of size k_1+k_2, tr $(\cdot)$ is the trace operator, and $\mathbf{L}_i$ includes both the local geometry and the discriminative information, which is given by

$$\mathbf{L}_i = \begin{pmatrix} -\mathbf{1}_{k_1+k_2}^T \\ \mathbf{I}_{k_1+k_2} \end{pmatrix} \operatorname{diag}(\mathbf{w}_i) (-\mathbf{1}_{k_1+k_2}, \mathbf{I}_{k_1+k_2}). \tag{9}$$

For classification, samples close to the classification boundary tend to be misclassified. Hence the samples close to classification boundary is more important for finding the subspace for classification. For considering the influence of these samples and evaluating the importance for solving the problem, margin degree m_i was proposed [57]. For the ith sample $\overline{\mathbf{p}}_i$, margin degree is defined as

$$m_i = \exp\left(-\frac{1}{(n_i+\delta)t} \right), \tag{10}$$

where n_i is the number of inter-class samples which fall in the given region around $\overline{\mathbf{p}}_i$, δ is a regularization parameter, and t is a scaling factor. The larger n_i is, the larger m_i will be. This means sample $\overline{\mathbf{p}}_i$ is much closer to the classification boundary. Margin degree is an effective measure for evaluating the importance of a sample in the whole objective function. If no inter-class sample lies around $\overline{\mathbf{p}}_i$, n_i is equal to zero and $m_i = \exp(-1/\delta t)$ is the minimum of margin degree m_i according to (10). To encode the importance of sample $\overline{\mathbf{p}}_i$, the part objective function (6) is weighted by the margin degree m_i. So we have the weighted part objective function:

$$\arg\min_{\mathbf{Z}_i} m_i \operatorname{tr}\left(\mathbf{Z}_i \mathbf{L}_i \mathbf{Z}_i^T \right) = \arg\min_{\mathbf{Z}_i} \operatorname{tr}\left(\mathbf{Z}_i m_i \mathbf{L}_i \mathbf{Z}_i^T \right). \tag{11}$$

For all samples, the whole objective function is combined with all weighted part objective function together linearly, i.e.,

$$\arg\min_{\mathbf{Z}_1,\ldots,\mathbf{Z}_{Nmn}} \sum_{i=1}^{Nmn} \operatorname{tr}\left(\mathbf{Z}_i m_i \mathbf{L}_i \mathbf{Z}_i^T \right). \tag{12}$$

We define a selection matrix:

$$(\mathbf{S}_i)_{pq} = \begin{cases} 1, & \text{if } p = S_i\{q\} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Utilizing the selection matrix $\mathbf{S}_i \in \mathbb{R}^{Nmn \times (k_1+k_2+1)}$, $\mathbf{Z}_i$ can be rewritten as

$$\mathbf{Z}_i = \mathbf{Z}\mathbf{S}_i, \tag{14}$$

where $\mathbf{Z}$ is the low-dimensional representation of $\mathbf{P}$, i.e., $\mathbf{Z} = \mathbf{U}^T\mathbf{P}$. Thus the whole objective function (12) is rewritten as

$$\begin{aligned}
&\underset{\mathbf{Z}}{\operatorname{argmin}} \sum_{i=1}^{Nmn} \operatorname{tr}(\mathbf{Z}\mathbf{S}_i m_i \mathbf{L}_i \mathbf{S}_i^T \mathbf{Z}^T) \\
&= \underset{\mathbf{Z}}{\operatorname{argmin}} \operatorname{tr}\left(\mathbf{Z} \left(\sum_{i=1}^{Nmn} \mathbf{S}_i m_i \mathbf{L}_i \mathbf{S}_i^T \right) \mathbf{Z}^T \right) \\
&= \underset{\mathbf{Z}}{\operatorname{argmin}} \operatorname{tr}\left(\mathbf{Z}\mathbf{L}\mathbf{Z}^T \right),
\end{aligned} \tag{15}$$

where $\mathbf{L} = \sum_{i=1}^{Nmn} \mathbf{S}_i m_i \mathbf{L}_i \mathbf{S}_i^T \in \mathbb{R}^{Nmn \times Nmn}$ is the alignment matrix.

Since DLA is a linear dimension reduction algorithm, we substitute $\mathbf{Z} = \mathbf{U}^T\mathbf{P}$ into the objective function (15) and have

$$\arg\min_{\mathbf{U}} \operatorname{tr}\left(\mathbf{U}^T\mathbf{P}\mathbf{L}\mathbf{P}^T\mathbf{U} \right) \text{ s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}_{D_1}. \tag{16}$$

Similar to PCA, DLA obtains an orthonormal projection matrix. Therefore we can solve the objective function (16) by using the standard eigen-value decomposition:

$$\mathbf{P}\mathbf{L}\mathbf{P}^T\mathbf{u} = \lambda\mathbf{u}. \tag{17}$$

The eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{D_1}$ are the solutions of (16) and the corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{D_1}$ are ordered in a descending order. Finally, we have the projection matrix $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{D_1})$.

In practice, PCA is suggested before DLA, because it can reduce noise in a certain degree [57]. Therefore the entire DLA will go through the following steps:

- Project the data $\mathbf{P}$ onto the subspace by utilizing PCA with the projection matrix $\mathbf{U}_{PCA}$.
- Compute the projection matrix $\mathbf{U}_{DLA}$ for the projective data $\mathbf{U}_{PCA}^T\mathbf{P}$ by employing DLA described previously.
- Achieve the final projection matrix through multiplying these two projection matrix, i.e.,

$$\overline{\mathbf{U}}^1 = \mathbf{U}_{PCA}\mathbf{U}_{DLA}. \tag{18}$$

In DLANet, the projection matrix forms the filter bank and the filters of DLANet are expressed as

$$\mathbf{K}_{d_1}^1 = \operatorname{mat}_{l,l}\left(\overline{\mathbf{u}}_{d_1}^1 \right), \; d_1 = 1,2,\ldots,D_1, \tag{19}$$

where $\operatorname{mat}_{l,l}(\cdot)$ reshapes the column vector in $\mathbb{R}^{l^2}$ to a matrix in $\mathbb{R}^{l \times l}$ and $\overline{\mathbf{u}}_{d_1}^1$ is the dth column of the projection matrix $\overline{\mathbf{U}}^1$ in Eq. (18). Given the filters, we have the outputs of this layer utilizing the convolution operator. The output maps are

$$\mathbf{X}_{i,d_1}^1 = \mathbf{X}_i * \mathbf{K}_{d_1}^1, \; d_1 = 1,2,\ldots,D_1, \tag{20}$$

where $*$ is the 2D convolution with the zero-padding to keep the size of output map being equal to that of the input image. For constructing a deeper network, the output maps can be fed to the second DLA layer.

3.2. The second DLA layer

The output maps of the first layer are regarded as input maps for the second layer. Hence we have ND_1 input maps $\{\mathbf{X}_{i,d_1}^1\}_{i,d_1=1,1}^{N,D_1}$ with labels derived from the original input scene images $\{\mathbf{X}_i\}_{i=1}^N$. The same as the first DLA layer, we collect the blocks from $\{\mathbf{X}_{i,d_1}^1\}_{i,d_1=1,1}^{N,D_1}$ and normalize them by subtracting their means. So we have the data matrix $\mathbf{P} \in \mathbb{R}^{l^2 \times D_1 Nmn}$ and then the filters $\mathbf{K}_{d_2}^2$, $d_2 = 1,2,\ldots,D_2$, are learnt by the method described in Section 3.1. For each input map $\mathbf{X}_{i,d_1}^1$, we convolves it with $\mathbf{K}_{d_2}^2$ and get the output maps of the second DLA layer, i.e.,

$$\mathbf{X}_{i,d_1,d_2}^2 = \mathbf{X}_{i,d_1}^1 * \mathbf{K}_{d_2}^2, \; d_2 = 1,2,\ldots D_2. \tag{21}$$

Totally we have D_1D_2 output maps in the second layer. They can be used as the input of the third DLA layer. Repeat the above process if more layers can boost the performance.

3.3. Feature layer

Unlike the traditional CNN flatting the original output maps as the feature, a special block-wise histogram feature which offers translation, rotation and scale invariance [56] is applied to the output maps of the last DLA layer.

Given the output maps of the ith image for the second DLA layer, we have D_2 maps $\mathbf{X}^2_{i,d_1,1}, \mathbf{X}^2_{i,d_1,2}, \ldots, \mathbf{X}^2_{i,d_1,}$, which are from the same input map $\mathbf{X}^1_{i,d_1}$. To binarize the output maps, we define:

$$B(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}, \tag{22}$$

which is an element-wise function for a matrix. The pixels at the same location are treated as a D_2 bits vector and then the vector is converted to a decimal number. Thus we get the output decimal map:

$$\mathbf{F}_{i,d_1} = \sum_{d_2=1}^{D_2} 2^{d_2-1} B(\mathbf{X}^2_{i,d_1,d_2}). \tag{23}$$

The range of elements in $\mathbf{F}_{i,d_1}$ is $[0,\ 2^{D_2}-1]$. After the binarization operation, the number of output maps of one scene image reduces from D_1D_2 to D_1. For each decimal map $\mathbf{F}_{i,d_1}$ of the ith image, we take the blocks for computing histogram. We compute the histogram with 2^{D_2} bins which are divided $[0,\ 2^{D_2}-1]$ into 2^{D_2} this in each block. Because we need the local descriptor, we concatenate D_1 histograms at the same position into a feature vector:

$$\mathbf{f}_i^{(x,y)} = \left(\mathrm{hist}\left(\mathbf{p}_{i,1}^{(x,y)}\right), \mathrm{hist}\left(\mathbf{p}_{i,2}^{(x,y)}\right), \ldots, \mathrm{hist}\left(\mathbf{p}_{i,D_1}^{(x,y)}\right)\right)^T, \tag{24}$$

where hist$(\cdot)$ is the histogram operator, $\mathbf{p}_{i,d_1}^{(x,y)} \in \mathbb{R}^{B_f \times B_f}$ is the block of size $B_f \times B_f$ taken at the coordinate (x, y) of the d_1 th decimal map $\mathbf{F}_{i,d_1}$ of the ith scene image. Therefore we have a local DLANet feature descriptor $\mathbf{f}_i^{(x,y)} \in \mathbb{R}^{D_1 \cdot 2^{D_2}}$, and its dimension depends on the number of filters D_1 and D_2. The dimensionality of the DLANet features grows up with D_2 exponentially but increases with D_1 linearly. In order to avoid extremely high computation and storage, we only vary the parameter D_1 to evaluate the influence of performance in experiments. Inspired by other local features, e.g., SIFT, HOG, we set the size of blocks $B_f = 16$ with stride $S_f = 8$, which means that half region of a block is overlapped by the next block.

Although the DLANet feature descriptors can be encoded by using LLC, LLC based on a given overcomplete basis requires that the codebook size is much larger than the dimension of the descriptors. If we set $D_1=8$ and $D_2=8$, the descriptor dimensionality is $8 \times 2^8 = 2048$. The general settings of the codebook size is less than or equal to 2048 for scene classification tasks [17,35], so we have to reduce the dimensionality of DLANet feature descriptors. PCA is a common method for dimension reduction without using the discriminative information. In addition, Ke and Sukthankar [65] applied PCA to SIFT. They demonstrated that the PCA–SIFT is more distinctive, more robust to image deformations, and more compact than the standard SIFT. In [56], whitening PCA (WPCA) is applied alternately for dimensionality reduction. PCA bases are weighted by the inverse of the corresponding square-root eigenvalues, i.e.,

$$\mathbf{U}_{WPCA} = \Lambda^{-1/2}\mathbf{U}_{PCA}, \tag{25}$$

where $\mathbf{\Lambda}^{-1/2} = \mathrm{diag}\,(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_D^{-1/2})$. The WPCA projection matrix $\mathbf{U}_{WPCA}$ equally treats variance along all principal component axes by weighting base vectors corresponding to smaller eigenvalues more heavily and may be appropriate for discrimination [66]. Thus, we can reduce the dimensionality of the descriptor $\mathbf{U}_{WPCA}^T\mathbf{f}_i^{(x,y)}$ and encode it by LLC–SPM.

4. Experimental results

In this section, we use the proposed DLANet features as the low level local feature under the LLC–SPM frame [19]. We evaluate the performance of the frame with the proposed DLANet features on three kinds of scene dataset, NYU Depth V1 [7], Scene-15 [25] and MIT Indoor-67 [26]. We also compare our method with some widely used hand-craft features, e.g., SIFT [34], HOG [36] and GIST [6], and learnable features, e.g., PCANet and LDANet [56].

We describe our experimental settings now. First, all images are convert to gray scale and various features are extracted in grayed RGB images for comparison. On NYU Depth V1, we also extract these features in depth images. For local features, local image block is set to 16×16 with stride 8, according to [35]. We will discuss later in detail. Second, we employ LLC–SPM to generate image representation. The codebook is computed by utilizing k-means clustering on local block descriptors randomly sampled from training set. The centers of k-means clustering algorithm are initialized through selecting k descriptors randomly. We terminate the iteration when it reaches the max iteration 100 or the centers change slightly. In our experiments, the codebook size is $k=1024$. SPM can pool the codes togeher to represent the images. Max pooling is highly recommended for LLC and three-layered spatial pyramid, 1×1, 2×2 and 4×4, is used. Hence the dimension of the representation of an image is $(1+2^2+4^2) \times 1024 = 21504$. Specially, we concatenate the representation of RGB images and depth images as the final representation of the RGBD image pairs on NYU Depth V1 dataset. Eventually, a linear SVM [67] is trained for classification, which is more suitable for LLC–SPM [19]. In addition, we use IFV [21] on Scene-15 and MIT Indoor-67. We employed GMM with 256 components and two-layered average pooling.

For comparison, we evaluate various features on the NYU Depth V1 dataset and the Scene-15 dataset, including SIFT, HOG, GIST, PCANet, LDANet and DLANet. We introduce them as follows.

SIFT: Dense SIFT descriptors are widely used. Typically a SIFT descriptor is extracted from a 16×16 block partitioned to 4×4 subregion. For each subregion, histogram with magnitude and orientation is computed with 8 bins. Thus we have a 128 dimensional feature vector formed by 4×4 histograms. To enhance the invariance to illumination, the feature vector is normalized.

HOG: The Histogram of Oriented Edges (HOG) descriptors are originally design for pedestrian detection. HOG is similar to SIFT. The histogram of magnitude and orientation are computed with 8 bins in the local image block. In [10], histograms from multiple HOG cells are stacked to provide more descriptive features which significantly improve the performance empirically. Inspired by their experiments, we generate multiple HOG cells by using three-layered spatial pyramid. We have pyramid HOG features of $(1+2^2+4^2) \times 8 = 168$ dimensions.

GIST: GIST uses Gabor-like filters with 8 orientations and 4 scales. The images are divided to 4×4 subregions. For each subregion, the responses of one orientation and one scale are averaged as the output. Therefore the representation of an image is an $8 \times 4 \times 16 = 512$ dimensional descriptor. Since GIST is a global feature, this descriptor is fed to linear SVM without applying LCC–SPM.

PCANet and LDANet [56]: PCANet and LDANet are multi-layer features which are simple and efficient. In the convolutional architecture, PCA and LDA are adopted to learn the projection matrix as filters. There is an extra output layer instead of nonlinearity and max pooling between tow convolutional layers in the network. The features are reduced by WPCA. The same as the settings in [56], the PCANet is trained with $D_1=D_2=8$ number of filters with size $l=7$. But the number of filters in LDANet is 6 because the reduced dimensionality must be less than the number of classes.

DLANet: The proposed DLANet is detailed in Section 3. The parameters of DLANet are set to $D_1=D_2=8$, $l=7$, which are the same as those of PCANet. For MIT Indoor-67, the local DLANet descriptors are extracted with size of blocks $B_f=8$ and stride $S_f=4$. Besides, the unique parameters $k_1=3$, $k_2=2$, $\gamma=0.05$ are empirically set for each DLA layer.

4.1. NYU depth V1

The NYU Depth V1 dataset, shown in Fig. 4, is collected by the New York University. Depth information which contains both geometric information and distance of objects are added into dataset. The depth images are acquired by Microsoft Kinect, which fulfilled the empty regions and smooth the noise by using the cross-bilateral filter because of the defect of the infra-red laser projector in Kinect. Totally, 2347 pairs of images are labeled and they can be grouped into seven categories, including bathroom, bedroom, bookstore, cafe, kitchen, living room, and office. Table 1 summarizes the dataset.

Following the common benchmarking procedures, we repeat the experimental process 10 times with randomly selected training samples (30 samples per scene category) and test samples to obtain reliable results. The training set is used to obtain learnable features, including PCANet, LDANet and DLANet. The average and standard deviation of the recognition rates are reported.

The mean accuracy and the standard deviation of different features on the NYU Depth V1 dataset are shown in Table 2 SIFT is the best feature among these three hand-craft features. Even though SIFT is designed for color images, it significantly surpass the HOG and GIST for depth images. It is reported that GIST is not suitable for indoor environments [1] and the representations derived from coding methods is better than the simple concatenation of local descriptors. Not surprisingly, GIST is much worse than the other features, even the mean accuracy for depth images is 49.43%. Obviously, the learnable features mostly outperform the hand-craft features. For color images, the accuracies of all learnable features are slightly higher than SIFT except LDANet which is not as good as PCANet reported in [56]. However, for depth images, LDANet feature also win the SIFT about 2.4%, which means that the features learned from depth images can adaptively fit the dataset.

It is notable that by concatenating color and depth representations, accuracies obtained by all features are improved. But only using depth images cannot achieve good performance. Hence we can use the depth information for improving scene classification. DLANet feature for RBGD data performs best, which is not only superior over the SIFT [56] but also outperforms PCANet and LDANet. We also note that the standard deviations of the performance of DLANet feature are relatively low. Hence it is an efficient and robust feature for RBGD data.

Table 1
Statistics of dataset.

Scene classes	Scenes	Number of samples
Bathroom	6	70
Bedroom	17	463
Bookstore	3	781
Cafe	1	47
Kitchen	10	276
Living Room	13	342
Office	14	305
Total	64	2347

Table 2
Classification accuracy on NYU Depth V1.

Feature	RGB	Depth	RGBD
SIFT [35]	78.1 ± 1.7	68.5 ± 1.5	79.9 ± 1.5
HOG [10]	73.54 ± 1.76	61.49 ± 1.85	76.12 ± 1.48
GIST [6]	63.53 ± 1.26	49.43 ± 2.43	70.13 ± 1.52
PCANet [56]	79.66 ± 1.54	$\mathbf{72.02} \pm 1.72$	81.59 ± 1.55
LDANet [56]	76.91 ± 1.67	70.91 ± 1.89	80.39 ± 1.73
DLANet	$\mathbf{80.33} \pm 1.51$	71.43 ± 1.56	$\mathbf{82.66} \pm 1.21$

Best results are displayed in bold.

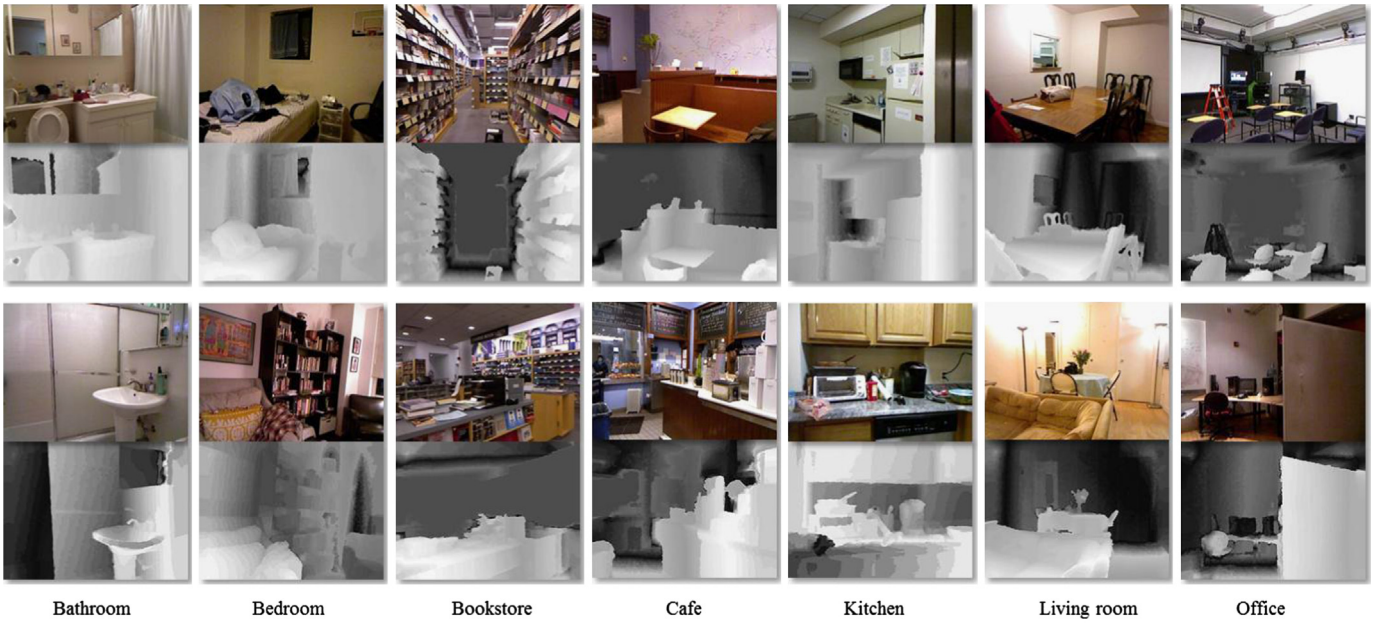

Fig. 4. Example images in the NYU Depth V1 dataset. We display 14 paired samples in seven indoor classes. Each pair has a color image and its corresponding depth image shown in gray scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Scene-15

The Scene-15 dataset contains 15 scene categories, which extends a dataset with 13 categories provided by Fei-Fei and Perona [68] by a dataset collected by Lazebnik et al. [25]. It consists of 4485 images of average size around 300×250. The number of images of each category varies between 215 and 410. All images are collected from the COREL collection, personal photographs, and Google image search. The Scene-15 dataset has both indoor and outdoor scenes, man-made and natural environments, unlike the NYU Depth V1 dataset introduced in the previous section. For scene classification task, Scene-15 is in common used for evaluating algorithms. Therefore we not only compare our proposed DLANet feature to other features as the previous section, but also compare the best result of our system to other methods, e.g., VQ–SPM [25], ScSPM [17], LLC–SPM [19], macrofeatures [23], FV–LRF [30] and KDES [70]. According to the suggestion in [25], we randomly select 100 images per class and repeat the training/testing for 10 times.

Table 3
Comparison on different features on Scene-15.

Feature	Accuracy
SIFT	82.26 ± 0.53
HOG	75.27 ± 0.69
GIST	66.61 ± 0.68
PCANet	82.73 ± 0.40
LDANet	84.75 ± 0.69
DLANet	$\mathbf{85.13} \pm 0.38$

Best results are displayed in bold.

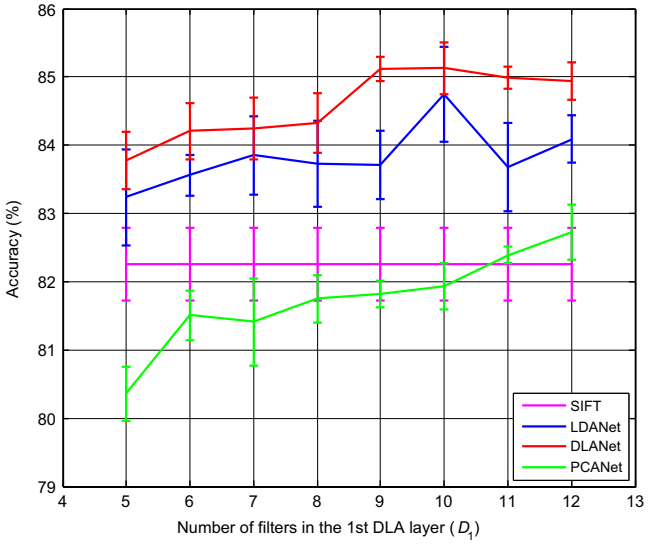

Fig. 5. Classification accuracy on the Scene-15 while varying the number of filters in the first DLA layer.

In Table 3, we discover a similar conclusion that the learned features outperform the hand-craft features in Section 4.1. PCANet feature is slightly superior to the hand-craft feature SIFT. However, by making use of the discriminative information, LDANet feature and DLANet feature outperform the PCANet feature. But the classification accuracy of LDANet feature for color images in Table 2 is much lower than the PCANet feature. This indicates that LDANet needs a large training set.

The results of three learnable features are the best results when the parameter D_1 is varied from 5 to 12. In Fig. 5, we use SIFT as the baseline whose accuracy is a constant because it is independent to the number of filters in the first DLA layer D_1. When $D_1 < 11$, the accuracy of PCANet feature is lower than SIFT. But it increases with the increasing of the number of filters in the first DLA layer D_1. However, both DLANet and LDANet features are superior to SIFT no matter what we set the parameter D_1. The DLANet feature achieves its highest accuracy 85.13% at $D_1 = 9$, while LDANet feature achieves its highest accuracy 84.75% at $D_1 = 10$. We also find that the standard deviations of DLANet feature is much lower than that of the other features.

We also study the sensitivity of the parameters k_1, k_2, γ in DLA layers. The other settings are fixed and k_1, k_2, γ are varied respectively. In Fig. 6(a) and (b), we vary k_1, k_2 from 2 to 6 and fix γ as 0.05. When we vary the parameter k_1 or k_2, the accuracy not change a lot. But there is a peak which corresponds to $k_1 = 3$ and $k_2 = 6$. In Fig. 6(c), we select γ from {0.01, 0.05, 0.1, 0.5, 1} and keep $k_1 = 3$, $k_2 = 2$. We find that the accuracy goes down when γ tends to be large. It is indicated that the parameters should be carefully selected.

We also evaluate the time for learning PCANet, LDANet and DLANet. We have used 12×2.4 GHz GPUs for evaluation. It took 6529.4 s to get all PCANet features of 4485 images. The run time of

Table 4
Comparison on different methods on Scene-15.

Method	Accuracy
VQ–SPM [25]	81.4 ± 0.5
ScSPM [17]	80.4 ± 0.45
LLC–SPM [19]	82.26 ± 0.53
Macrofeatures [23]	84.3 ± 0.5
KDES [69]	81.9 ± 0.6
FV–LRF [30]	85.0 ± 0.6
ISPR [31]	85.08 ± 0.01
GPP [32]	85.13 ± 0.72
ISPR+IFV [31]	$\mathbf{91.06} \pm 0.05$
PCANet [56]	82.73 ± 0.40
LDANet [56]	84.75 ± 0.69
DLANet	85.13 ± 0.38
DLANet+IFV	90.14 ± 0.21

Best results are displayed in bold.

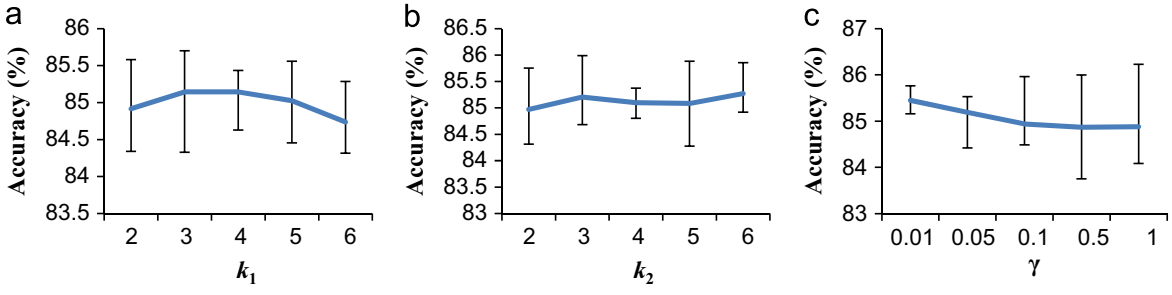

Fig. 6. Classification accuracy on the Scene-15 while varying the DLA parameters k_1, k_2, γ.

Table 5
Comparison on different methods on MIT Indoor-67.

Method	Accuracy
Object Bank [70]	37.6
VQ+SPM [71]	34.4
OC Kernels [72]	39.85
GPP [32]	46.38
ISPR [31]	50.10
MC+OBJPOOL [73]	55.9
FV–LRF [30]	60.3
ISPR+IFV [31]	**68.5**
DLANet	46.27
DLANet+IFV	59.10

Best results are displayed in bold.

LDANet and DLANet are 6499.0 s and 6879.4 s respectively. The main difference between PCANet, LDANet and DLANet is in the filter learning stages. For PCANet and LDANet, we should compute covariance matrix and its eigenvectors. For DLANet, we have to find top k_1 nearest intra-class neighbors and top k_2 nearest inter-class neighbors in the whole sample space of each samples for constructing the alignment matrix firstly. Although we use k–d tree instead of the exhaustive search method, the computation of DLANet is still slightly higher than PCANet and LDANet.

Since the Scene-15 dataset is widely-used, we compare the proposed DLANet feature based LLC–SPM with other methods which all belong to the encoding framework. The codebook sizes of all methods are set equally to 1024 for fair comparison. As shown in Table 4, DLANet feature outperforms all the other methods. ScSPM, LLC–SPM and macro features utilize sparse coding or its variation in encoding stage. Macro features group the neighbor SIFT descriptors as a local feature getting a fairly high accuracy 84.3%, which is also a method for modifying the low level feature. KDES can be considered as a learnable feature, but it learns the feature in kernel space. Note that we reference its result achieve by combining color, gradient and shape kernel descriptors. FV–LRF is different to other method, which apply Fisher vector encoding instead of sparse coding and changing pooling region adaptively. Similarly, both ISPR [31] and GPP [32] focus on modifying the spatial pooling method. GPP further encodes the LLC by the Geometric Phrase Pooling [32] algorithm which calculates a codeword from itself and its neighbors. In [32], early fusion features of SIFT and Edge-SIFT are used to capture texture and shape features together. The weighted spatial pooling is also proposed to filter the noises from the descriptors not on the objects that we want to recognize. ISPR model [31] can jointly learn the important spatial pooling regions and the appearances. The DLANet achieves almost the same accuracy of GPP and slightly higher than ISPR. However, as ISPR is an excellent complementarity to IFV, the accuracy of the combination of DLANet+IFV is lower than ISPR+IFV.

4.3. MIT Indoor-67

The MIT Indoor-67 dataset [26] contains 67 indoor scenes and totally 15,620 images. We used the original splits in [26], which used 80 images per class for training and 20 images for testing. For learning the features efficiently, we resized the images to be no bigger than 300×300 pixels with preserved aspect ratio.

In Table 5, we show that the accuracies of different methods vary from 37.6% to 68.5%. The proposed method is slightly worse than GPP [32] but have a large gap of ISPR [31] and MC+OBJPOOL [73]. After fusing the fisher vector, FV–LRF [30] and ISPR+IFV [31] achieve much higher accuracies. Similarly, the performance of DLANet is improved by fusing the fisher vector, which is close to FV–LRF [30], but inferior to ISPR+IFV.

5. Conclusion

In this paper, we presented a manifold learning-based discriminative feature learning network DLANet for scene classification. Inspired by PCANet [56], we learn the filter banks by applying Discriminative Locality Alignment (DLA) which follows the manifold assumption. In a local region of a given sample, DLA pushes the inter-class neighbor samples away and narrow the intra-class neighbor samples in projected space. For all samples, margin degree is explored to measure the importance of the corresponding sample for classification. Through combining these two objectives, we conduct the DLA which can extract the discriminative features for classification. By utilizing the block-wise histograms of the binary codes, we obtain efficient and robust local descriptors. Employing those DLANet features, we construct the scene classification system under the LLC–SPM framework.

To verify the effectiveness of DLANet, we evaluate it on the NYU Depth V1 dataset, the Scene-15 dataset and the MIT Indoor-67 dataset. We compare the proposed DLANet with the hand-craft features, e.g., SIFT, HOG, and GIST, and learnable features, PCANet and LDANet. Not surprisingly, almost all learnable features outperform the hand-craft features on color images, but they overwhelmingly won the hand-craft features on depth images. In addition, we examine the influence of the parameter setting of PCANet, LDANet and DLANet and achieve the highest accuracy when choosing an appropriate D_1. Experimental results show that PCANet, LDANet and DLANet always outperform SIFT. Finally, we compare the DLANet feature combined with the LLC–SPM scheme with other methods and show DLANet with LLC–SPM is competitive to other approaches. Therefore, DLANet feature is an efficient and robust learnable feature for scene classification.

Acknowledgments

This research is supported in part by NSFC, China (Grant no.: 61075021, 61201348, 61472144), National Science and Technology Support plan (Grant no.: 2013BAH65F01-2013BAH65F04), GDNSF (Grant no.: S2011020000541, S2012040008016, S2013010014240), GDSTP (Grant no.: 2012A010701001), Research Fund for the Doctoral Program of Higher Education of China (Grant no.: 20120172110023), Opening Project of State Key Laboratory of Digital Publishing Technology, Shenzhen Technology Project (JCYJ20140901003939001).

References

[1] J. Wu, M. Rehg, CENTRIST: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.
[2] A. Ude, R. Dillmann, Vision-based robot path planning, Adv. Robot Kinemat. Comput. Geom. (1994) 505–512.
[3] A. Pronobis, B. Caputo, P. Jensfelt, H.I. Christensen, A realistic benchmark for visual indoor place recognition, Robot. Auton. Syst. 58 (1) (2010) 81–96.
[4] A. Bosch, A. Zisserman, X. Muñoz, Scene classification using a hybrid generative/discriminative approach, IEEE Trans. Pattern Anal. Mach. Intell. 30 (4) (2008) 712–727.
[5] Vogel Julia, Bernt Schiele, Semantic modeling of natural scenes for content-based image retrieval, Int. J. Comput. Vis. 72 (2) (2007) 133–157.
[6] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[7] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: Proceedings of ICCV Workshop 3-D Representation and Recognition, November 2011, pp. 601–608.
[8] N. Silberman, D. Hoiem, P. Kohli, R. Fergus, Indoor segmentation and support inference from RGBD images, in: Proceedings of European Conference on Computer Vision, 2012.
[9] J. Xiao, A. Owens, A. Torralba, Sun3d: a database of big spaces reconstructed using SFM and Object labels, in: ICCV, 2013.
[10] J. Xiao, K.A. Ehinger, J. Hays, A. Torralba, A. Oliva, SUN database: exploring a large collection of scene categories, Int. J. Comput. Vis. (2014) 1–20.
[11] A. Bosch, X. Muñoz, R. Marti, A review: which is the best way to organize/classify images by content, Image Vis. Comput. 25 (6) (2007) 778–791.
[12] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.

[13] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of ACM International Conference on Multimedia, 1996, pp. 65–73.
[14] F. Mindru, T. Tuytelaars, L. Van Gool, T. Moons, Moment invariants for recognition under changing viewpoint and illumination, Proc. Comput. Vis. Image Understand. 94 (1–3) (2004) 3–27.
[15] A. Vailaya, A. Figueiredo, A. Jain, H. Zhang, Image classification for content-based indexing, IEEE Trans. Image Process. 10 (2001) 117–129.
[16] J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, A.W.M. Smeulders, Kernel codebooks for scene categorization, in: Proceedings of European Conference on Computer Vision, 2008.
[17] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2009, pp. 1794–1801.
[18] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Proceedings of Information and Processing Systems, 2009.
[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of IEEE International Conference on Computer Vision Pattern Recognition, June 2010, pp. 3360–3367.
[20] F. Perronnin, C.R. Dance, Fisher kernels on visual vocabularies for image categorization, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2007, pp. 1.
[21] F. Perronnin, J. Sanchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: IEEE Conference on Computer Vision Pattern Recognition, 2010.
[22] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proceedings of BMVC, 2011.
[23] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: IEEE Conference on Computer Vision Pattern Recognition, 2010, pp. 2559–2566.
[24] Z. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2011, pp. 1697–1704.
[25] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, June 2006, pp. 2167–2178.
[26] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2009, pp. 413–420.
[27] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric l_p-norm feature pooling for image classification, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2011.
[28] Boureau, Y-Lan, Jean Ponce, Yann LeCun, A theoretical analysis of feature pooling in visual recognition, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010.
[29] Y. Jia, C. Huang, T. Darrell, Beyond spatial pyramids: receptive field learning for pooled image features, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2012, pp. 3370–3377.
[30] C. Xu, N. Vasconcelos, Learning receptive fields for pooling from tensors of feature response, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2014.
[31] D. Lin, C. Lu, R. Liao, J. Jia, Learning important spatial pooling regions for scene classification, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2014.
[32] L. Xie, Q. Tian, M. Wang, B. Zhang, Spatial pooling of heterogeneous features for image classification, IEEE Trans. Image Process. 23 (5) (2014) 1994–2008.
[33] L. Xie, J. Wang, B. Guo, B. Zhang, Q. Tian, Orientational pyramid matching for recognizing indoor scenes, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2014.
[34] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[35] D. Tao, L. Jin, Z. Yang, X. Li, Rank preserving sparse learning for kinect based scene classification, IEEE Trans. Cybern. 43 (5) (2013) 1406–1417.
[36] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2005, pp. 886–893.
[37] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[38] S. Tang, X. Wang, X. Lv, X. Han, J. Keller, Z. He, M. Skubic, S. Lao Histogram of oriented normal vectors for object recognition with a depth sensor, in: Computer Vision – ACCV 2012, 2013, pp. 525–538.
[39] L. Bo, X. Ren, D. Fox, Depth kernel descriptors for object recognition, in: Intelligent Robots and Systems (IROS), September 2011, pp. 821–826.
[40] G.E. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[41] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[42] G.E. Hinton, R.S. Zemel, Autoencoders, minimum description length, and Helmholtz free energy, in: Proceedings of Neural Information and Processing Systems, 1993.
[43] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: Proceedings of International Conference on Machine Learning, 2011.
[44] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of International Conference on Machine Learning, 2008.
[45] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[46] D.H. Hubel, T.N. Wiesel, Receptive fields of single neurons in the cat's striate cortex, J. Physiol. 148 (1959) 574–591.
[47] A. Coates, A.Y. Ng, The importance of encoding versus training with sparse coding and vector quantization, in: Proceedings of International Conference on Machine Learning, 2011.
[48] M. Norouzi, M. Ranjbar, G. Mori, Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2009, pp. 2735–2742.
[49] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of International Conference on Machine Learning, 2009.
[50] M. Henaff, K. Jarrett, K. Kavukcuoglu, Y. LeCun, Unsupervised learning of sparse features for scalable audio classification, in: Proceedings of International Conference on Music Information Retrieva, 2011.
[51] M. Zeiler, D. Krishnan, G. Taylor,R. Fergus, Deconvolutional networks, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2010.
[52] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of Neural Information and Processing Systems, 2012.
[53] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Proceedings of Neural Information and Processing Systems, 2012, pp. 3642–3649.
[54] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1915–1929.
[55] Y. Sun, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, arXiv preprint arXiv:1406.4773, 2014.
[56] T.H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma, PCANet: a simple deep learning baseline for image classification? arXiv preprint arXiv:1404.3606, 2014.
[57] T. Zhang, D. Tao, J. Yang Discriminative locality alignment, in: Proceedings of European Conference on Computer Vision, pp. 725–738, 2008.
[58] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[59] J. Tenenbaum, V. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[60] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1299–1313.
[61] N. Guan, D. Tao, Z. Luo, B. Yuan, Non-negative patch alignment framework, IEEE Trans. Neural Netw. 22 (8) (2011) 1218–1230.
[62] T. Zhou, D. Tao, X. Wu, Manifold elastic net: a unified framework for sparse dimension reduction, Data Min. Knowl. Disc 22 (3) (2011) 340–371.
[63] Y. Mu, D. Tao, X. Li, Biologically inspired tensor features, Cogn. Comput. 1 (4) (2009) 327–341.
[64] L. Zhang, L. Zhang, D. Tao, X. Huang, Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction, IEEE Trans. Geosci. Remote Sens. 51 (1) (2013) 242–256.
[65] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2004.
[66] H.V. Nguyen, L. Bai, L. Shen, Local gabor binary pattern whitened PCA: a novel approach for face recognition from single image per person, Adv. Biom. 5558 (2009) 269–278.
[67] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines [Online]. Available: ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩, 2001.
[68] L. Fei-Fei, P. Perona A Bayesian hierarchical model for learning natural scene categories, in: Proceedings of IEEE Conference on Computer Vision Pattern Recognition, 2005.
[69] L. Bo, X. Ren, D. Fox, Kernel descriptors for visual recognition, in: Proceedings of Neural Information and Processing Systems, 2010, pp. 244–252.
[70] L. -J. Li, H. Su, Y. Lim, L. Fei-Fei, Object bank: a high-level image representation for scene classification & semantic feature sparsification, in: Proceedings of Neural Information and Processing Systems, 2010, pp. 1378–1386.
[71] M. Pandey, S. Lazebnik, Scene recognition and weakly supervised object localization with deformable part-based models, in: ICCV, 2011, pp. 1307–1314.
[72] L. Zhang, X. Zhen, L. Shao, Learning object-to-class kernels for scene classification, IEEE Trans. Image Process. 23 (8) (2014) 3241–3253.
[73] A. Bergamo, L. Torresani, Classemes and other classifier-based features for efficient object categorization, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 1988–2001.

Ziyong Feng received the B.S. degree from the College of Engineering at South China Agricultural University, Guangzhou, China. He is currently pursuing the Ph.D. degree in information and communication engin;eering at the South China University of Technology, Guangzhou, China.

His current research interests include machine learning, computer vision.

Lianwen Jin (M'98) received the B.S. degree from the University of Science and Technology of China, Anhui, China, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 1991 and 1996, respectively.

He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. He is the author of more than 100 scientific papers. His current research interests include image processing, handwriting analysis and recognition, machine learning, cloud computing, and intelligent systems.

Dr. Jin is a member of the China Image and Graphics Society and the Cloud Computing Experts Committee of the China Institute of Communications. He was a recipient of the award of New Century Excellent Talent Program of MOE in 2006 and the Guangdong Pearl River Distinguished Professor Award in 2011. He served as a Program Committee member for a number of international conferences, including ICMLC2007～2011, ICFHR2008-2012, ICDAR2009, ICDAR2013, ICPR2010, ICPR2012, ICMLA2012, etc.

His research interests include image processing, handwriting analysis and recognition, machine learning, cloud computing, and intelligent systems.

Dapeng Tao received the B.S. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China. He is currently pursuing the Ph.D. degree in information and communication engineering at the South China University of Technology, Guangzhou, China.

His current research interests include machine learning, computer vision, and cloud computing.

Shuangping Huang received M.S. degree sand Ph.D. degree from the South China University of Technology in 2005 and 2011, respectively. She is currently a lecturer in the College of Engineering at South China Agricultural University, Guangzhou, China.

Her research interests include machine learning, computer vision and data mining.